# Transforming acoustic characteristics to deceive playback spoofing countermeasures of speaker verification systems

Fuming Fang, Junichi Yamagishi, Isao Echizen, Md Sahidullah, Tomi Kinnunen

# Transforming acoustic characteristics to deceive playback spoofing countermeasures of speaker verification systems

Fuming Fang[1], Junichi Yamagishi[1,2], Isao Echizen[1], Md Sahidullah[3], Tomi Kinnunen[4]
[1]National Institute of Informatics, Japan [2]University of Edinburgh, UK
[3]Inria, France [4]University of Eastern Finland, Finland
{fang, jyamagis, iechizen}@nii.ac.jp, md.sahidullah@inria.fr, tkinnu@cs.uef.fi

## Abstract

*Automatic speaker verification (ASV) systems use a playback detector to filter out playback attacks and ensure verification reliability. Since current playback detection models are almost always trained using genuine and played-back speech, it may be possible to degrade their performance by transforming the acoustic characteristics of the played-back speech close to that of the genuine speech. One way to do this is to enhance speech "stolen" from the target speaker before playback. We tested the effectiveness of a playback attack using this method by using the speech enhancement generative adversarial network to transform acoustic characteristics. Experimental results showed that use of this "enhanced stolen speech" method significantly increases the equal error rates for the baseline used in the ASVspoof 2017 challenge and for a light convolutional neural network-based method. The results also showed that its use degrades the performance of a Gaussian mixture model-universal background model-based ASV system. This type of attack is thus an urgent problem needing to be solved.*

## 1. Introduction

Automatic speaker verification (ASV) [1], a kind of biometrics authentication technology, identifies a person from a segment of speech. ASV systems typically fall into two types: text-independent and text-dependent, where the latter requests a client to speak a given phrase. Due to the convenience of ASV, it is being used in more and more applications, such as ones used in call centers and by mobile devices. However, ASV is vulnerable to several kinds of spoofing attacks (also known as presentation attacks [2]), so ASV systems need a spoofing countermeasure (CM) (also known as presentation attack detection [2]). Such attacks aim to mimic the target speaker mainly by using synthesized speech [3], converted speech [3], or playback speech [4, 5]. Among them, playback speech-based attacks are relatively easy to mount since an attacker who has no special knowledge can make them [6]. Once an attacker has collected/stolen a voice sample for the target speaker, he/she can simply play it back to an ASV system or concatenate segments of the sample to form a new utterance. Threats from this kind of attack have been confirmed by several studies [4, 5, 7, 8, 9]. Here we focus on playback spoofing attacks and relevant CMs.

Four main types of CMs have been developed to protect against playback spoofing attacks. *One type* utilizes a text-dependent ASV system and randomly prompts for a pass-phrase [10, 11], making it difficult to mount playback attacks using phrase-fixed speech. However, it is possible to form an arbitrary utterance to spoof this type of CM if the attacker has sufficient speech data for the target speaker. *The second type* is based on rules describing the characteristics of genuine speech (recorded from a person). For example, Mochizuki et al. [12] distinguished genuine speech by detecting pop-noise from certain phonemes. An intractable problem related to this type of CM is that it is difficult to design suitable rules and implement them. *The third type* utilizes audio fingerprinting to check whether an incoming recording is similar to previously authenticated utterances that were automatically saved in the ASV system. Rodriguez et al. [13] developed such a system: if the similarity score was higher than a threshold, the recording was treated as a playback attack. A disadvantage of this type of CM is that it is sensitive to noise. In contrast, *the fourth type* compares the differences between genuine speech and playback speech. This type mainly utilizes a machine learning algorithm to learn the differences. An example is Wang et al.'s [14] use of a support vector machine [15] to learn the difference in Mel-frequency cepstral coefficient (MFCC)-based acoustic features.

More methods of the fourth type were presented at the second Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2017), in which a common database was used to assess the participants' CMs. The database consists of two parts. One part contains genuine speech taken from the RedDots corpus [16], which was

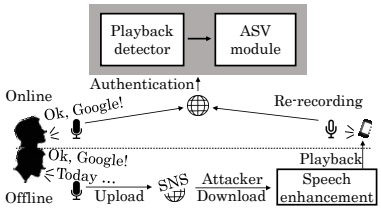


Figure 1. Playback spoofing attack using enhanced stolen speech method under ASVspoof 2017 scenario. Without speech enhancement, attack is the same as a conventional playback attack.

designed for speaker verification. The other part contains recordings of the genuine speech made in various environments. For these data, the baseline [4] with a constant Q cepstral coefficient (CQCC) [17] feature and a Gaussian mixture model (GMM) classifier had an equal error rate (EER) of 30.60%. A deep learning-based method had an EER of 6.73% [18], which was the best performance achieved at ASVspoof 2017 [4].

These mainstream CMs of the fourth type are also problematic: they are based on the assumption that the attackers do not have special knowledge. Moreover, this type of CM algorithms only learn the difference from a given dataset and perhaps do not work well if the acoustic characteristics of the playback speech is transformed close to that of the genuine one. To confirm this hypothesis, we tested the effectiveness of a playback attack using speech "stolen" from the target speaker and enhanced before mounting the attack. This enhancement should remove the distortions in the stolen speech caused by the recording device and environmental noise so that they do not affect the re-recorded speech.

We evaluated the effectiveness of a playback attack using this enhanced stolen speech method against a text-dependent ASV system. We used the ASVspoof 2017 scenario (Figure 1) in which the attacker is assumed to obtain from somewhere uncompressed speech for the target speaker containing the phrase used for authentication, e.g., by downloading from the web, hacking a device used by the target speaker, and talking to and surreptitiously recording the target speaker. The speech enhancement generative adversarial network (SEGAN) [19] was used to transform the acoustic characteristics of the obtained speech close to that of the genuine speech. We also investigated the effect of different types of playback loudspeakers and re-recording devices. The results showed that it is possible to fool playback spoofing CMs by transforming the acoustic characteristic of the playback speech close to that of the genuine speech.

## 2. Related work

Pioneering work on playback attacks was reported by Lindberg and Blomberg in 1999 [20]. They pre-recorded the numbers one to ten of two speakers and then concatenated various combinations of them to attack a hidden Markov model (HMM) [21]-based text-dependent ASV system. They demonstrated considerable increase in both the EER and false acceptance rate (FAR) compared to verification without attacks. More recently, Ergunay et al. investigated the effect of playback attacks against ASV systems and also achieved a large increase in FAR [22]. Compared to these conventional playback attacks, our method further degrades the performance of playback spoofing CMs by enhancing the speech.

There are a few attack methods similar to our enhanced stolen speech method. Demiroglu et al. improved the naturalness of synthesized and converted speech before attacking a phase-based synthetic speech detector and an ASV system [23]. The synthesized and converted speech signals were firstly analyzed frame by frame, and each frame was replaced with one containing the most similar natural speech selected from a dataset. A complex cepstrum vocoder was used to re-synthesize these frames so as to improve speech naturalness. Finally, the speech was directly fed into an ASV system. They reported that their method fooled four out of nine detectors. Our method can be thought of as an extension of their method as it further transforms synthesized speech close to natural speech.

Nguyen et al. reported an attack method that transforms computer-generated (CG) images into natural images before feeding them into a facial authentication system [24]. The transformation model is trained using a generative adversarial network (GAN) [25]. The GAN discriminator, which mimics a spoofing detector, is used to distinguish CG/natural images. The discriminator is pre-trained and fixed during training of the transformation model. In contrast, we treat the authentication system as a black box, and anything regarding playback spoofing CMs and ASV systems is unknown.

## 3. Playback detectors and ASV system

Two playback spoofing CMs and a classical Gaussian mixture model with universal background model (GMM-UBM) [26]-based ASV system were used to evaluate the effectiveness of our enhanced stolen speech attack method. The two CMs were the baseline and the core method of the system with the best performance (i.e., a light convolutional neural network) of ASVspoof 2017.

### 3.1. Baseline of ASVspoof 2017

The baseline of ASVspoof 2017 consists of a CQCC front-end and a GMM back-end. We refer to this method as "CQCC-GMM CM". The CQCC is an acoustic feature extracted from an audio signal. CQCC extraction is performed using constant Q transform (CQT) instead of the classical short-time Fourier transformation (STFT). STFT suffers from fixed frequency resolution and fixed temporal resolution whereas CQT exhibits higher frequency resolution at lower frequencies and higher temporal resolution at higher frequencies. An audio signal is usually represented

by a sequence of CQCC feature vectors.

A two-class GMM-based classifier is used for genuine/playback speech detection. One GMM is trained using genuine speech while the other one is trained using playback speech. Input to the GMMs is CQCC-based acoustic feature vectors, and the expectation maximization (EM) [27] algorithm is used for training. During prediction, the feature vectors of an audio signal are independently input into the two models, and then the joint log-likelihood for both models is calculated. Finally, the log-likelihood ratio of the genuine and playback model results is compared with a threshold to determine genuine/playback speech.

### 3.2. Core method of best system of ASVspoof 2017

The best system of ASVspoof 2017 was a fusion of three sub-systems: a support vector machine with $i$-vector features [28], a convolutional neural network (CNN) with a recurrent neural network (RNN), and a light CNN (LCNN). The LCNN was used as the core method, which achieved an EER of 7.37%. This performance was very close to that of the fused system (6.73%). We therefore used an "LCNN CM" to evaluate our enhanced stolen speech attack method.

The LCNN consists of five convolution layers, four network-in-network (NIN) [29] layers, ten max-feature-map (MFM) layers, five max-pooling layers, and two fully connected layers. Each MFM layer acts as a maxout activation function [30] that splits the CNN feature maps into two groups and then performs element-wise maximization to select features. The LCNN input is a spectrum with a fixed size of $864 \times 400$, which is obtained by performing STFT with 1728 bins and concatenating 400 frames. Dropout [31] is applied after the first fully connected layer. The final output layer, with a softmax activation function, is used to discriminate genuine/playback speech. This is described in more detail elsewhere [18].

The silence parts of the audio signal are removed, and then STFT is performed using a window length of 25 ms with a shift size of 10 ms. If a signal is shorter than 4 seconds, its content is repeated to match the length. For a longer signal, its content is repeated to match multiples of 4 seconds, and the output probabilities are averaged.

### 3.3. GMM-UBM-based ASV system

We use a GMM-UBM-based system for speaker verification. Even though it is a classic ASV method, GMM-UBM provides competitive performance on short-duration, text-dependent ASV tasks [32]. The speaker models are created by maximum a posteriori adaptation from a UBM trained with a large amount of speech data from different speakers. Text-dependent speaker models are separately created for different passphrases following the guidelines for conducting experiments with the RedDots corpus. The recognition score is the likelihood ratio between the results of the target speaker model and those of the UBM.

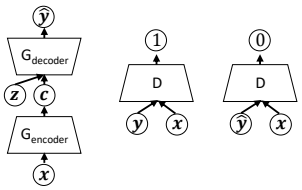


Figure 2. Architecture of SEGAN. Input of discriminator is $(\mathbf{y}, \mathbf{x})$ or $(\hat{\mathbf{y}}, \mathbf{x})$; the former should be classified as real data while the latter should be classified as fake data; but the latter should be treated as real data when updating the generator parameters, so adversarial training is performed.

## 4. Speech enhancement

SEGAN is a data-driven speech enhancement method that constructs a mapping from a noisy waveform to a clean waveform with the help of supervised training. More specifically, SEGAN leverages the power of the GAN composed of two adversarial networks, a discriminator $D$ and a generator $G$. The discriminator predicts the probability that the input is from real data rather than from fake data generated by $G$. The generator learns a mapping function from a prior noise distribution $p_{\text{noise}}(\mathbf{z})$ to the distribution of the real data $p_{\text{data}}(\mathbf{y})$ to fool the discriminator. If the noise distribution is conditioned by $\mathbf{x}$ drawn from playback speech and $\mathbf{y}$ drawn from genuine speech, the output $\hat{\mathbf{y}}$ is genuine-like speech.

The objective function for training SEGAN is formulated as

$$\begin{aligned}\min_D \mathcal{L}(D) &= \frac{1}{2}\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y})}[(D(\mathbf{y},\mathbf{x})-1)^2] \\ &\quad + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\text{noise}}(\mathbf{z})}[D(\hat{\mathbf{y}},\mathbf{x})^2] \\ \min_G \mathcal{L}(G) &= \mathbb{E}_{\mathbf{z}\sim p_{\text{noise}}(\mathbf{z})}[(D(\hat{\mathbf{y}},\mathbf{x})-1)^2] \\ &\quad + \lambda \cdot \| \hat{\mathbf{y}} - \mathbf{y} \|_1, \end{aligned} \tag{1}$$

where $\hat{\mathbf{y}} = G(\mathbf{z}, \mathbf{x})$ is enhanced (or generated) speech and $\mathbb{E}$ means expectation. $L_1$ norm, $\| \cdot \|_1$, is used to measure the distance between the real and enhanced speech. The discriminator and generator are alternately trained by performing a min-max game.

Figure 2 shows the architecture of SEGAN. It is an end-to-end model, so both the input and output of the generator are raw waveforms. The input of the discriminator is also a raw waveform combining $(\mathbf{y}, \mathbf{x})$ or $(\hat{\mathbf{y}}, \mathbf{x})$. The generator has an encoder-decoder structure. The encoder part is composed of 11 stacked 1-D CNN layers with a filter width of 31 and a stride of two. The decoder part is a mirror structure of the encoder part, and the corresponding layers between them are connected by a skip path. The dimension of noise $\mathbf{z}$ is the same as that of encoder output $\mathbf{c}$ and is drawn from a normal distribution. They are concatenated and input to the decoder. The architecture of the discriminator is the same as that of the encoder part of the generator except that virtual batch-normalization [33] is performed in hidden layers.

## 5. Database

In this section, we describe the speech data used for training the spoofing detectors, ASV, and SEGAN. We also describe the test data used for authentic as well as illegitimate access.

### 5.1. Training data for playback spoofing CM

Both the CQCC-GMM and the LCNN CM models were trained using the ASVspoof 2017 database (version 2), which was derived from the RedDots corpus. The genuine speech data in the database was taken from the RedDots corpus, and the playback speech data was recorded by playing the genuine speech data in various environments (including quiet and noisy places) using various recording devices and speakers with various qualities. The sampling rate was 16 kHz for both the genuine and playback speech. The database was further split into three datasets: training, development, and evaluation. Each of the datasets contained both genuine and playback speech. The GMM of the CQCC-GMM CM was trained using the training and development datasets. For the LCNN, the training dataset was used to estimate the model parameters and the development dataset was used to monitor the training process.

### 5.2. Training data for ASV system

We used TIMIT and RSR2015 (background subset [34]) corpora for training the UBM for the GMM-UBM-based ASV system. Only male speakers were used as the ASVspoof 2017 database was created from the male subset of the RedDots corpus. In total, we used 17,850 speech utterances from 488 speakers for UBM training. Each target speaker model was created with speech utterances from three different sessions for a fixed passphrase.

### 5.3. Training data for SEGAN

We used a high-quality database and two low-quality databases distorted by recording devices or environment noises to train SEGAN. The high-quality database was the voice cloning toolkit (VCTK) corpus [35]. This corpus contains data recorded in a hemi-anechoic chamber by 109 native English speakers, but we used data for only 28 speakers. One of the low-quality databases was a device-recorded VCTK (DR-VCTK) corpus [36] and the other one was a noisy VCTK (N-VCTK) corpus [37]. The DR-VCTK was created by playing the high-quality speech of the 28 speakers in office environments and recording it using relatively inexpensive consumer devices. The N-VCTK was created by adding noise to the high-quality speech of the 28 speakers. The sampling rate of these databases was 48 kHz with downsampling to 16 kHz. Two types of SEGAN were trained. One was trained using DR-VCTK and VCTK. The other was trained using N-VCTK and VCTK.

### 5.4. Authentication and spoofing data

We equally split the genuine speech of the evaluation dataset in the ASVspoof 2017 database into two subdatasets. One was used as authentication speech, and the other was used as "stolen speech." We enhanced the stolen speech, played it using four types of portable loudspeakers, and re-recorded it using six types of recording devices in an office room. The four loudspeakers were a high-quality speaker (BOSE Soundlink), a medium-quality speaker (SONY SRS-BTS50), a low-quality speaker (audio-technica AT-SP92), and an iPhone 6s speaker. The six recording devices were a high-quality condenser microphone (Apogee MiC 96k), a directional microphone (Sony ECM–673), a low-quality microphone (Snowball iCE), a MacBook microphone, an iPad microphone, and an iPhone 6s microphone. These devices were placed at around 30 to 50 cm from the loudspeaker. The sampling rate for the re-recording was 16 kHz. According to the used loudspeaker, four playback and re-recording sessions were performed.

## 6. Experimental setup

We evaluated 1) how our enhanced stolen speech method affects the performance of playback spoofing CMs and 2) how the enhanced speech effects an ASV system. In order to compare the difference between our method and conventional playback attacks, a paired two-tailed $t$-test was used.

### 6.1. Setup for playback spoofing CMs

Settings of the CQCC-GMM CM were the same as baseline of ASVspoof 2017 and the source code is available at [38]. CQCC had 29 dimension and 0-th order cepstral coefficient was further used. Their first and second derivatives were finally used as features (90 dimensions in total). The GMM of the CQCC-GMM CM had 512 components.

The weights of the LCNN were initialized using the Xavier method [39]. The dropout rate was set to 0.5. Adam optimization [40] with momentum of 0.5 was used. The initial learning rate was 0.0001; it was reduced by 0.9 if the classification accuracy of the development dataset decreased after each epoch. There were nine epochs, and the mini-batch size was 64. The LCNN was implemented using the TensorFlow framework [41] and is available at [42].

To assess the performance of both spoofing CMs, we use EER, which reflects the ability of the CM to discriminate genuine speech samples from playback attacks.

### 6.2. Setup for ASV system

Our ASV system used MFCC-based acoustic feature extracted from a 20 ms short-term window with a 10 ms shift using 20 filters. We computed 19 MFCCs after discarding the energy coefficients. The MFCCs were further processed with RASTA filtering to suppress convolutive mismatch. The delta and double-delta coefficients were computed for a context of three frames and then augmented with static MFCCs to create a 57-dimensional feature vector. Finally, cepstral mean and variance normalization (CMVN) was performed after discarding the non-speech frames with an energy-based voice activity detector. We trained the gender-dependent UBM with 512 mixture components. The

Table 1. EERs for CQCC-GMM CM. Bold means largest degradation.

| Loudspeaker used for replay | Enhancement training data | Directional microphone | High-quality microphone | Low-quality microphone | Mac book | iPad | iPhone 6s | Average |
|---|---|---|---|---|---|---|---|---|
| High quality | – | 15.65 | 8.83 | 20.87 | 9.98 | 7.21 | 49.92 | 18.74 |
| | DR-VCTK | 28.42 | 18.10 | 29.67 | 14.96 | 8.59 | 50.00 | 24.96 |
| | N-VCTK | 35.61 | **23.18** | 33.59 | 16.49 | 9.17 | 50.00 | 28.01 |
| Medium quality | – | 9.35 | 11.96 | 8.78 | 10.78 | 6.54 | 49.13 | 16.09 |
| | DR-VCTK | 15.71 | 20.16 | 15.76 | 15.27 | 7.43 | 49.92 | 20.71 |
| | N-VCTK | 22.56 | 25.62 | 22.03 | 15.61 | 8.36 | 49.92 | 24.02 |
| Low quality | – | 11.83 | 8.98 | 10.28 | 8.34 | 6.14 | 49.92 | 15.92 |
| | DR-VCTK | 20.07 | 16.29 | 19.77 | 10.32 | 6.96 | 49.96 | 20.56 |
| | N-VCTK | 26.87 | 22.35 | 24.44 | 10.78 | 7.35 | 49.92 | 23.62 |
| iPhone 6s | – | 16.28 | 16.54 | 19.83 | 7.19 | 6.40 | 49.53 | 19.30 |
| | DR-VCTK | 30.45 | 31.19 | 30.50 | 10.93 | 7.14 | 49.92 | 26.69 |
| | N-VCTK | 24.25 | 24.26 | 26.94 | 9.83 | 7.28 | 49.88 | 23.74 |

Table 2. EERs for LCNN CM. Bold means largest degradation.

| Loudspeaker used for replay | Enhancement training data | Directional microphone | High-quality microphone | Low-quality microphone | Mac book | iPad | iPhone 6s | Average |
|---|---|---|---|---|---|---|---|---|
| High quality | – | 11.19 | 8.00 | 16.14 | 7.71 | 12.95 | 25.04 | 13.51 |
| | DR-VCTK | 12.35 | 9.12 | 18.14 | 8.59 | 13.55 | 25.74 | 14.58 |
| | N-VCTK | 13.48 | 10.43 | 19.74 | 8.85 | 13.83 | 25.29 | 15.27 |
| Medium quality | – | 8.78 | 9.98 | 5.92 | 5.47 | 7.09 | 25.25 | 10.42 |
| | DR-VCTK | 9.57 | 11.22 | 6.56 | 6.85 | 8.79 | 27.25 | 11.71 |
| | N-VCTK | 10.31 | 12.22 | 7.96 | 7.23 | 9.56 | 27.12 | 12.40 |
| Low quality | – | 7.25 | 6.06 | 5.31 | 9.52 | 7.80 | 16.29 | 8.71 |
| | DR-VCTK | 8.44 | 7.10 | 6.07 | 10.08 | 8.76 | 17.05 | 9.58 |
| | Noisy VCTK | 10.23 | **8.95** | 7.52 | 10.30 | 9.38 | 17.09 | 10.58 |
| iPhone 6s | – | 11.11 | 11.56 | 10.47 | 4.40 | 9.17 | 17.65 | 10.73 |
| | DR-VCTK | 13.25 | 14.97 | 13.62 | 5.23 | 11.12 | 18.26 | 12.74 |
| | N-VCTK | 11.70 | 12.33 | 11.21 | 4.54 | 10.07 | 18.07 | 11.32 |

speaker models were created by adapting only the centers of UBM with a relevance factor of three.

Even though ASV spoofing evaluations have focused on standalone CM assessment, the performance of a tandem (combined) system is important for real-world deployment. Both CM and ASV can result in target speaker misses and false acceptances of impostors (either non-targets or spoofs). We therefore adopted a recently proposed *tandem detection cost function* (t-DCF) metric [43] for evaluating the combination of two systems in a Bayes risk framework. The t-DCF is given by $C_{\text{miss}}^{\text{asv}} \cdot \pi_{\text{tar}} \cdot P_a + C_{\text{fa}}^{\text{asv}} \cdot \pi_{\text{non}} \cdot P_b + C_{\text{fa}}^{\text{cm}} \cdot \pi_{\text{spoof}} \cdot P_c + C_{\text{miss}}^{\text{cm}} \cdot \pi_{\text{tar}} \cdot P_d$, where $C_{\text{miss}}^{\text{asv}} = 1$, $C_{\text{fa}}^{\text{asv}} = 10$, $C_{\text{fa}}^{\text{cm}} = 10$, and $C_{\text{miss}}^{\text{cm}} = 1$ are unit costs related to the misses and false alarms of the two systems; $\pi_{\text{spoof}} = 0.0100$, $\pi_{\text{non}} = 0.0099$, and $\pi_{\text{tar}} = 0.9801$ were the prior probabilities of the targets, non-targets, and spoofs, respectively; and $P_a$, $P_b$, $P_c$, and $P_d$ are the error rates of four possible errors originating from the joint actions of the CM and ASV systems. The reported t-DCF values are minimum t-DCF values with a fixed ASV system. The higher the value, the less usable the combined (ASV and CM) system.

### 6.3. Setup for SEGAN

Similar to previous work [19, 44], we extracted chunks of waveforms by using a sliding window of $2^{14}$ samples at every $2^{13}$ samples (i.e., 50% overlap). During testing, we concatenated the results at the end of the stream without overlap. The learning rate, mini-batch size, and epoch size were set to 0.0002, 100, and 120, respectively. The $\lambda$ in Equation 1 was set to 100. We used source code for improved SEGAN [45].

## 7. Results

Tables 1 and 2 show the EERs for the CQCC-GMM CM and the LCNN CM, respectively. Playback spoofing attacks using our enhanced stolen speech method had significantly higher EERs for both CMs compared to those of conventional playback attacks (without enhancement). One reason could be that the signal-to-noise ratio was higher after speech enhancement, resulting in the playing of cleaner speech. Use of the high-quality speaker with the high-quality microphone and use of the low-quality speaker with the high-quality microphone when N-VCTK was used to train SEGAN resulted in the largest performance degrada-

Table 3. Values of t-DCF obtained from combination of CQCC-GMM CM and ASV scores. Bold means largest degradation.

| Loudspeaker used for replay | Enhancement training data | Directional microphone | High-quality microphone | Low-quality microphone | Mac book | iPad | iPhone 6s | Average |
|---|---|---|---|---|---|---|---|---|
| High quality | – | 0.9361 | 0.9276 | 0.9392 | 0.9136 | 0.9118 | 0.9426 | 0.9285 |
| | DR-VCTK | 0.9412 | 0.9363 | 0.9410 | 0.9258 | 0.9210 | 0.9426 | 0.9347 |
| | N-VCTK | 0.9403 | 0.9373 | 0.9402 | 0.9277 | 0.9211 | 0.9431 | 0.9350 |
| Medium quality | – | 0.9156 | 0.9351 | 0.9211 | 0.9222 | 0.9039 | 0.9425 | 0.9234 |
| | DR-VCTK | 0.9313 | 0.9386 | 0.9348 | 0.9311 | 0.9143 | 0.9428 | 0.9322 |
| | N-VCTK | 0.9339 | 0.9377 | 0.9362 | 0.9308 | 0.9186 | 0.9429 | 0.9334 |
| Low quality | – | 0.9225 | 0.9177 | 0.9248 | 0.9109 | 0.9020 | 0.9415 | 0.9199 |
| | DR-VCTK | 0.9339 | 0.9314 | 0.9362 | 0.9220 | 0.9076 | 0.9428 | 0.9290 |
| | N-VCTK | 0.9353 | 0.9342 | 0.9363 | 0.9223 | 0.9105 | 0.9425 | 0.9302 |
| iPhone 6s | – | 0.9354 | 0.9358 | 0.9379 | 0.9085 | 0.9006 | 0.9384 | 0.9261 |
| | DR-VCTK | 0.9380 | 0.9379 | 0.9388 | **0.9290** | 0.9127 | 0.9381 | 0.9324 |
| | N-VCTK | 0.9387 | 0.9391 | 0.9383 | 0.9247 | 0.9100 | 0.9388 | 0.9316 |

Table 4. Values of t-DCF obtained from combination of LCNN CM and ASV scores. Bold means largest degradation.

| Loudspeaker used for replay | Enhancement training data | Directional microphone | High-quality microphone | Low-quality microphone | Mac book | iPad | iPhone 6s | Average |
|---|---|---|---|---|---|---|---|---|
| High quality | – | 0.9239 | 0.9073 | 0.9436 | 0.9063 | 0.9338 | 0.9656 | 0.9301 |
| | DR-VCTK | 0.9303 | 0.9135 | 0.9494 | 0.9098 | 0.9385 | 0.9664 | 0.9347 |
| | N-VCTK | 0.9346 | 0.9186 | 0.9522 | 0.9105 | 0.9390 | 0.9658 | 0.9368 |
| Medium quality | – | 0.9105 | 0.9173 | 0.8990 | 0.8978 | 0.9038 | 0.9657 | 0.9157 |
| | DR-VCTK | 0.9159 | 0.9266 | 0.9011 | 0.9021 | 0.9107 | 0.9673 | 0.9206 |
| | N-VCTK | 0.9202 | 0.9293 | 0.9077 | 0.9035 | 0.9143 | 0.9675 | 0.9238 |
| Low quality | – | 0.9043 | 0.8992 | 0.8972 | 0.9122 | 0.9070 | 0.9584 | 0.9131 |
| | DR-VCTK | 0.9105 | 0.9029 | 0.8999 | 0.9156 | 0.9125 | 0.9605 | 0.9170 |
| | N-VCTK | 0.9183 | 0.9108 | 0.9051 | 0.9156 | 0.9141 | 0.9593 | 0.9205 |
| iPhone 6s | – | 0.9208 | 0.9238 | 0.9173 | 0.8952 | 0.9120 | 0.9530 | 0.9204 |
| | DR-VCTK | 0.9344 | 0.9396 | **0.9345** | 0.8970 | 0.9222 | 0.9540 | 0.9303 |
| | N-VCTK | 0.9234 | 0.9274 | 0.9214 | 0.8960 | 0.9154 | 0.9559 | 0.9233 |

tion for the two CMs. The increases in EER were 2.6 and 1.5 times, respectively.

As expected, use of the high-quality speaker resulted in higher EERs because it generated more natural speech. It is interesting that the results for the iPhone 6s speaker were similar to those for the high-quality speaker. While a wide range of EERs were obtained for the recording devices, use of the high-quality microphone did not result in significantly higher EERs. The CQCC-GMM CM could not distinguish the playback speech re-recorded using the iPhone 6s. This was because features were not normalized and channel distortions greatly degraded its performance [5]. Enhancement based on the N-VCTK was mightier than that based on the DR-VCTK in most cases. This might be because distortion due to environmental noise has a greater effect than that due to the recording devices.

Tables 3 and 4 show the t-DCF values for the combined CQCC-GMM CM and GMM-UBM-based ASV scores and for the combined LCNN CM and GMM-UBM-based ASV scores, respectively. Compared to the conventional playback attacks, an attack using our enhanced stolen speech method greatly degraded the authentication performance of both combinations. This suggests that our enhanced stolen speech method enables playback attacks to pass playback spoofing CMs and to fool ASV systems as well.

## 8. Conclusion and future work

We investigated the effectiveness of using enhanced stolen speech in playback spoofing attacks. Experimental results showed that stolen speech enhanced with SEGAN can greatly degrade the performances of baseline CQCC-GMM and advanced LCNN-based playback spoofing CMs as well as that of GMM-UBM-based ASV systems.

Since the used speech enhancement method for attack would be unknown, we plan to develop a robust playback detection method for various speech enhancement methods.

## Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Numbers JP16H06302, 18H04120, 18H04112, 18KT0051, 17H04687, and Academy of Finland (project no. 309629). We thank Huy H. Nguyen, the Graduate University for Advanced Studies (SOKENDAI), for comments on an earlier versions of the manuscript.

## References


[1] J. H. Hansen and T. Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, 32(6):74–99, 2015. 1

[2] I. 30107-1:2016. Information technology Biometric presentation attack detection Part 1: Framework. https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-1:ed-1:v1:en, 2016. [Online; accessed 5-July-2018]. 1

[3] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 1

[4] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In *INTERSPEECH 2017, Annual Conference of the International Speech Communication Association, August 20-24, 2017, Stockholm, Sweden*, Stockholm, SWEDEN, 08 2017. 1, 2

[5] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi. ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements. In *ODYSSEY 2018, The Speaker and Language Recognition Workshop, June 26-29, 2018, Les Sables d'Olonne, France*, Les Sables d'Olonne, FRANCE, 06 2018. 1, 6

[6] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153, 2015. 1

[7] F. Alegre, A. Janicki, and N. Evans. Re-assessing the threat of replay spoofing attacks against automatic speaker verification. In *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the*, pages 1–6. IEEE, 2014. 1

[8] J. Gałka, M. Grzywacz, and R. Samborski. Playback attack detection for text-dependent speaker verification over telephone channels. *Speech Communication*, 67:143–153, 2015. 1

[9] Z. Wu, S. Gao, E. S. Cling, and H. Li. A study on replay attack and anti-spoofing for text-dependent speaker verification. In *APSIPA*, pages 1–5. IEEE, 2014. 1

[10] T. Kinnunen, M. Sahidullah, I. Kukanov, H. Delgado, M. Todisco, A. Sarkar, N. Thomsen, V. Hautamaki, N. Evans, and Z.-H. Tan. Utterance verification for text-dependent speaker recognition: a comparative assessment using the RedDots corpus. In *INTERSPEECH 2016, Annual Conference of the International Speech Communication Association, September 8-12, 2016, San Francisco, USA*, San Francisco, UNITED STATES, 09 2016. 1

[11] H. Zeinali, L. Burget, H. Sameti, and H. Cernocky. Spoken pass-phrase verification in the i-vector space. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 372–377, 2018. 1

[12] S. Mochizuki, S. Shiota, and H. Kiya. Voice liveness detection using phoneme-based pop-noise detector for speaker verification. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 233–239. 1

[13] J. Gonzalez-Rodriguez, A. Escudero, D. de Benito-Gorron, B. Labrador, and J. Franco-Pedroso. An audio fingerprinting approach to replay attack detection on asvspoof 2017 challenge data. In *Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 304–311. ISCA, 2018. 1

[14] C. Wang, Y. Zou, S. Liu, W. Shi, and W. Zheng. An efficient learning based smartphone playback attack detection using gmm supervector. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pages 385–389. IEEE, 2016. 1

[15] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 1

[16] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. v. Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, et al. The reddots data collection for speaker recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 1

[17] M. Todisco, H. Delgado, and N. Evans. A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients. In *Speaker Odyssey Workshop, Bilbao, Spain*, volume 25, pages 249–252, 2016. 2

[18] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin. Audio replay attack detection with deep learning frameworks. *Interspeech*, 2017. 2, 3

[19] S. Pascual, A. Bonafonte, and J. Serra. Segan: Speech enhancement generative adversarial network. In *Interspeech*. ISCA, 2017. 2, 5

[20] J. Lindberg and M. Blomberg. Vulnerability in speaker verification-a study of technical impostor techniques. In *Sixth European Conference on Speech Communication and Technology*, 1999. 2

[21] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 2

[22] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel. On the vulnerability of speaker verification to realistic voice spoofing. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–6. IEEE, 2015. 2

[23] C. Demiroglu, O. Buyuk, A. Khodabakhsh, and R. Maia. Postprocessing synthetic speech with a complex cepstrum vocoder for spoofing phase-based synthetic speech detectors. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):671–683, 2017. 2

[24] H. H. Nguyen, N.-D. T. Tieu, H.-Q. Nguyen-Son, J. Yamagishi, and I. Echizen. Transformation on computer-generated facial image to avoid detection by spoofing detector. *arXiv preprint arXiv:1804.04418*, 2018. 2

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[26] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000. 2

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 3

[28] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011. 3

[29] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 3

[30] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013. 3

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[32] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan. Further optimisations of constant q cepstral processing for integrated utterance and text-dependent speaker verification. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 179–185. IEEE, 2016. 3

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 3

[34] A. Larcher, K. A. Lee, B. Ma, and H. Li. Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication*, 60:56–77, 2014. 4

[35] C. Veaux, J. Yamagishi, and K. MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. http://dx.doi.org/10.7488/ds/1994, 2017. 4

[36] S. S. Sarfjoo and J. Yamagishi. Device recorded vctk (small subset version). http://dx.doi.org/10.7488/ds/2316, 2018. 4

[37] C. Valentini-Botinhao. Noisy speech database for training speech enhancement algorithms and TTS models. http://dx.doi.org/10.7488/ds/2117, 2017. 4

[38] ASVspoof 2017 Organizers. Baseline replay attack detector. http://www.asvspoof.org/data2017/baseline_CM.zip, 2017. 4

[39] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 4

[40] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 4

[42] F. Fang. A TensorFlow implementation of light convolutional neural network (LCNN). https://github.com/fangfm/lcnn, 2018. 4

[43] T. Kinnunen, K.-A. Lee, H. Delgado, N. W. D. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds. t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. *CoRR*, abs/1804.09618, 2018. 5

[44] J. Lorenzo-Trueba, F. Fang, X. Wang, I. Echizen, J. Yamagishi, and T. Kinnunen. Can we steal your vocal identity from the internet?: Initial investigation of cloning obamas voice using gan, wavenet and low-quality found data. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 240–247, 2018. 5

[45] S. S. Sarfjoo. Improved SEGAN. https://github.com/ssarfjoo/improvedsegan, 2017. 5